Report

This implementation of Cluedo has four packages into which the classes are split:

- Control - Contains everything that controls things on the board (Player and Dice)
- Game - Contains classes/objects that relate to the game itself, including the Cards and Board
- Locations - Contains all the implementations of the Location class (i.e. all of the locations on the board)
- Tests - Contains all the test classes for testing that the game works.

The Cluedo class is one of the main classes of the entire program. It holds references to all the important objects, like the board, players, cards (which are split up into their different types), dice and solution. It initialises the entire game by creating the characters, rooms, weapons, board, players and dealing the cards. It then runs a loop (controlled by a boolean ‘running’) which steps through all the players and takes their turns. The player must keep track of what they think the solution is themselves (just like in the real game).

Another main class is the Board. This contains the layout of the board, players and their positions, rooms, stairwells and the middle. The board layout is based off the board given in the handout. The addLocations() method links the Locations that can be moved to from each location.

The Player class is another main class which represents a player and their hand of cards, along with an associated character. The move() method rolls the dice and attempts to move the player using attemptMove(), which asks for a coordinate where the first character is a capital letter and the second and third (if there is a third) characters are digits. It then checks that the move is valid by traversing through the validMoves in the Location.

Location is the last main class. It is abstract, but contains the validMoves, which is a collection of Locations where the player can move to from the current location. Each implementation of the Location has a different way of working this out. The class also has a method called addLocation(), which is essentially a ‘dummy method’ that must be implemented in each different type of the Location, but actualAddLocation() is the method that actually adds it to the collection, otherwise there would have been a fair amount of code reuse.

A key object used in the code is Point, which is useful when working with 2D arrays, as they contain an x and y coordinate, meaning that a Point can represent a location in the 2D array quite simply.
The movement is implemented by the user selecting a coordinate to move to. If the player cannot move there, it asks again.
The card selection is implemented by the program printing out the possible choices and prompting the user to enter the corresponding number to their selection.
These ideas are easily to implement and get the job done.